



Universidad
Carlos III de Madrid



This is a postprint version of the following published document:

Griol D., Molina J.M., Sanchis de Miguel A. (2014) Domain and Subtask-Adaptive Conversational Agents to Provide an Enhanced Human-Agent Interaction. In: Demazeau Y., Zambonelli F., Corchado J.M., Bajo J. (eds) Advances in Practical Applications Heterogeneous Multi-Agent Systems. The PAAMS Collection. PAAMS 2014. Lecture Notes in Computer Science, vol 8473. pp 134-145 Springer, Cham

DOI: [10.1007/978-3-319-07551-8_12](https://doi.org/10.1007/978-3-319-07551-8_12)

Domain and Subtask-Adaptive Conversational Agents to Provide an Enhanced Human-Agent Interaction

David Griol, José Manuel Molina, and Araceli Sanchis de Miguel

Computer Science Department

Carlos III University of Madrid

Avda. de la Universidad, 30, 28911 - Leganés, Spain

{david.griol,josemanuel.molina,araceli.sanchis}@uc3m.es

Abstract. One of the most demanding tasks when developing conversational agents consists of designing the dialog manager, which decides the next system response considering the user's actions and the dialog history. A previously developed statistical dialog management technique is adapted in this work to reduce the effort and time required to design the dialog manager. This technique allows not only an easy adaptation to new domains, but also to deal with the different subtasks by means of specific dialog models adapted to each dialog objective in the domain of a multiagent system. The practical application of the proposed technique to develop a conversational agent providing railway information shows that the use of these specific dialog models increases the quality and number of successful interactions with the agent in comparison with developing a single dialog model for the complete domain.

Keywords: Human-agent interaction, User interfaces, Conversational agents, Speech interaction, Information systems, Statistical methodologies.

1 Introduction

Spoken conversational agents or dialog systems are computer programs that receive speech as input and generate as output synthesized speech, engaging the user in a dialog that aims to be similar to that between humans [1, 2]. Thus, these interfaces make technologies more usable, as they ease interaction [3], allow integration in different environments [4], facilitate the interaction with multiagent systems [5, 6] and make technologies more accessible, especially for disabled people [7].

Usually, conversational agents carry out five main tasks: Automatic Speech Recognition (ASR), Spoken Language Understanding (SLU), Dialog Management (DM), Natural Language Generation (NLG), and Text-To-Speech Synthesis (TTS). These tasks are typically implemented in different modules of the system's architecture.

The goal of speech recognition is to obtain the sequence of words uttered by a speaker. It is a very complex task, as there can be a great deal of variation in the input the recognizer must analyze, for example, in terms of the linguistics of the utterance, inter and intra speaker variation, the interaction context and the transmission channel. Once the speech recognizer has provided an output, the system must understand what the user said. The goal of spoken language understanding is to obtain the semantics from the recognized sentence. This process generally requires morphological, lexical, syntactical, semantic, discourse and pragmatical knowledge.

The dialog manager decides the next action of the system, interpreting the incoming semantic representation of the user input in the context of the dialog. In addition, it resolves ellipsis and anaphora, evaluates the relevance and completeness of user requests, identifies and recovers from recognition and understanding errors, retrieves information from data repositories, and decides about the next system's response. Natural language generation is the process of obtaining sentences in natural language from the non-linguistic, internal representation of information handled by the dialog system. Finally, the TTS module transforms the generated sentences into synthesized speech.

In order to enable rapid deployment of these agents, markup languages such as VoiceXML¹ have been widely adopted as they reduce the time and effort required for system implementation. However, typically hand-crafted dialog management strategies using rules and heuristics still involve a very costly engineering cycle in the system development with this approach [8]. As an attempt to reduce this cost and carry out rapid system prototyping, statistical approaches are gaining increasing interest [9–11].

Statistical approaches enable automatic learning of dialog strategies, thus avoiding the time-consuming process that hand-crafted dialog design involves. Statistical models can be trained from real dialogs, modeling the variability in user behaviors. Although the construction and parameterization of these models depend on expert knowledge about the interaction domain, the objective is to develop systems that are more robust for real-world conditions, and easier to adapt to different users and tasks [9].

The most widespread methodology for machine-learning of dialog strategies consists of modeling human-computer interaction as an optimization problem using Partially Observable Markov Decision Processes (MDP) and reinforcement methods [12]. The main drawback of this approach is that the large state space of practical domains makes its direct representation intractable [13].

In this paper we adapt a statistical approach for the development of dialog managers [10], which is mainly based on the use of a classification process for the estimation of a statistical model from the sequences of the system and user dialog acts obtained from a set of training data. This technique has been previously applied to develop dialog managers for domains of different complexity [10]. This paper is specially focused on the adaptation and evaluation of this technique when specific dialog models are learned for each dialog subtask instead of learning

a single dialog model for the complete conversational agent. To do this, the training data is divided into different subsets, each covering a specific dialog objective or subtask. We propose to use our approach to learn specific dialog models for each dialog subset instead of using the complete training data to learn a single dialog model for the task. These specific dialog models are selected by the dialog manager once the objective of the dialog has been detected, using the generic dialog model until this condition has been fulfilled.

We have applied the proposed methodology to develop two versions of a conversational agent providing railway information in Spanish. The first one uses a generic dialog model and the second one uses specific dialog models for each dialog objective. An in-depth comparative assessment of the conversational agents has been completed using both real users and a user-agent simulation technique recently developed [14]. The results of the evaluation show that the specific dialog models allow a better selection of the next system responses, thus increasing the number and quality of successful interactions with the agent.

The rest of the paper is organized as follows. Section 2 describes our proposal for developing statistical dialog managers with specific dialog models. Section 3 shows a practical implementation of our proposal to generate a specific system. In Section 4 we discuss the evaluation results obtained by comparing two baseline versions of the system with a context-aware version that adapts its behavior integrating our proposal. Finally, in Section 5 we present the conclusions and outline guidelines for future work.

2 Our Proposed Methodology for Dialog Management

This section summarizes the proposed dialog management technique and the practical implementation proposed in this paper by means of specific dialog models for each subtask.

2.1 Proposed Statistical Methodology

In order to control the interactions with the user, our dialog manager represents dialogs as a sequence of pairs (A_i, U_i) , where A_i is the output of the dialog manager (the system answer) at time i , and U_i is the semantic representation of the user turn (the result of the understanding process of the user input) at time i ; both expressed in terms of dialog acts [10]. Each dialog is represented by:

$$(A_1, U_1), \dots, (A_i, U_i), \dots, (A_n, U_n)$$

where A_1 is the greeting turn of the system, and U_n is the last user turn. We refer to a pair (A_i, U_i) as S_i , the state of the dialog sequence at time i .

In this framework, we consider that, at time i , the objective of the dialog manager is to find the best system answer A_i . This selection is a local process for each time i and takes into account the previous history of the dialog, that is to say, the sequence of states of the dialog preceding time i :

$$\hat{A}_i = \operatorname{argmax}_{A_i \in \mathcal{A}} P(A_i | S_1, \dots, S_{i-1}) \quad (1)$$

where set \mathcal{A} contains all the possible system answers.

Following Equation 1, the dialog manager selects the following system prompt by taking into account the sequence of previous pairs (A_i, U_i) . The main problem to resolve this equation is regarding the number of possible sequences of states, which is usually very large. To solve the problem, we define a data structure in order to establish a partition in this space, i.e., in the history of the dialog preceding time i . This data structure, which we call *Dialog Register (DR)*, contains the information provided by the user throughout the previous history of the dialog. After establishing the equivalence relation in the histories of dialogs, the selection of the best A_i is given by:

$$\hat{A}_i = \operatorname{argmax}_{A_i \in \mathcal{A}} P(A_i | DR_{i-1}, S_{i-1}) \quad (2)$$

Each user turn supplies the system with information about the task; i.e., the user asks for a specific concept and/or provides specific values for certain attributes. However, a user turn can also provide other kinds of information, such as task-independent information (for instance, *Acceptance*, *Rejection*, and *Not-Understood* dialog acts). This kind of information implies some decisions which are different from simply updating the DR_{i-1} . Hence, for the selection of the best system response A_i , we take into account the DR that results from turn 1 to turn $i - 1$, and we explicitly consider the last state S_{i-1} .

As stated before, the DR contains information about concepts and values for the attributes provided by the user throughout the previous history of the dialog. For the dialog manager to determine the next answer, we have assumed that the exact values of the attributes are not significant. They are important for accessing databases and for constructing the output sentences of the system. However, the only information necessary to predict the next action by the system is the presence or absence of concepts and attributes. Therefore, the codification we use for each field in the DR is in terms of three values, $\{0, 1, 2\}$, according to the following criteria: (0) The concept is unknown or the value of the attribute is not given; (1) the concept or attribute is known with a confidence score that is higher than a given threshold; (2) the concept or attribute has a confidence score that is lower than the given threshold. To decide whether the state of a certain value in the DR is 1 or 2, the system employs confidence measures provided by the ASR and SLU modules.

2.2 Proposed Implementation by Means of Specific Dialog Models

As a practical implementation of this methodology, in this paper we propose the use of two modules. The first module deals with the detection of the specific dialog objective described by the user. This detection is based on the specific semantic information regarded to the task that is provided by the SLU module. This module also updates the *Dialog Register* that contains the complete list

of features provided by the SLU module through the dialog history until the current moment. Until a specific problem is detected, a generic model learned with all the training dialogs is used for the selection of the next system response.

Once the objective of the dialog has been detected, a second module uses a specific dialog model learned for each subtask to select the next system response. To do this, we propose to solve Equation 2 by means of a classification process. This way, every dialog situation (i.e., each possible sequence of dialog acts) is classified taking into account a set of classes \mathcal{C} , in which a class contains all the sequences that provide the same set of system actions (responses). The objective of the dialog manager at each moment is to select a class of this set $c \in \mathcal{C}$, so that the system response is the one associated with the selected class.

The classification function can be defined in several ways. We have previously evaluated six different definitions of such a function: a multinomial naive Bayes classifier, an n-gram based classifier, a decision tree classifier, a support vector machine classifier, a classifier based on grammatical inference techniques, and a classifier based on artificial neural networks [10]. The best results were obtained using a multilayer perceptron (MLP) [15] where the input layer holds the input pair (DR_{i-1}, S_{i-1}) corresponding to the dialog register and the state. The values of the output layer can be seen as an approximation of the a posteriori probability of the input belonging to the associated class $c \in \mathcal{C}$. Figure 1 shows the described scheme for the practical implementation of the proposed dialog management technique and its interaction with the rest of the modules in the conversational agent.

3 Practical Application

Within the framework of the DIHANA project, a mixed-initiative conversational agent was developed to provide a railway information system using spontaneous speech in Spanish [16]. The system integrates the CMU Sphinx-II system speech recognition module². As in many other conversational agents, the semantic representation chosen for dialog acts of the SLU module is based on the concept of frame [17]. This way, one or more concepts represent the intention of the utterance, and a sequence of attribute-value pairs contains the information about the values given by the user. For the task, we defined eight concepts and ten attributes. The eight concepts are divided into two groups:

1. *Task-Dependent Concepts*: they represent the concepts the user can ask for (*Timetables*, *Fares*, *Train-Type*, *Trip-Time*, and *Services*).
2. *Task-Independent Concepts*: they represent typical interactions in a dialog (*Acceptance*, *Rejection*, and *Not-Understood*).

The attributes are: *Origin*, *Destination*, *Departure-Date*, *Arrival-Date*, *Class*, *Departure-Hour*, *Arrival-Hour*, *Train-Type*, *Order-Number*, and *Services*. A total of 51 responses were defined for the system, corresponding to the request of

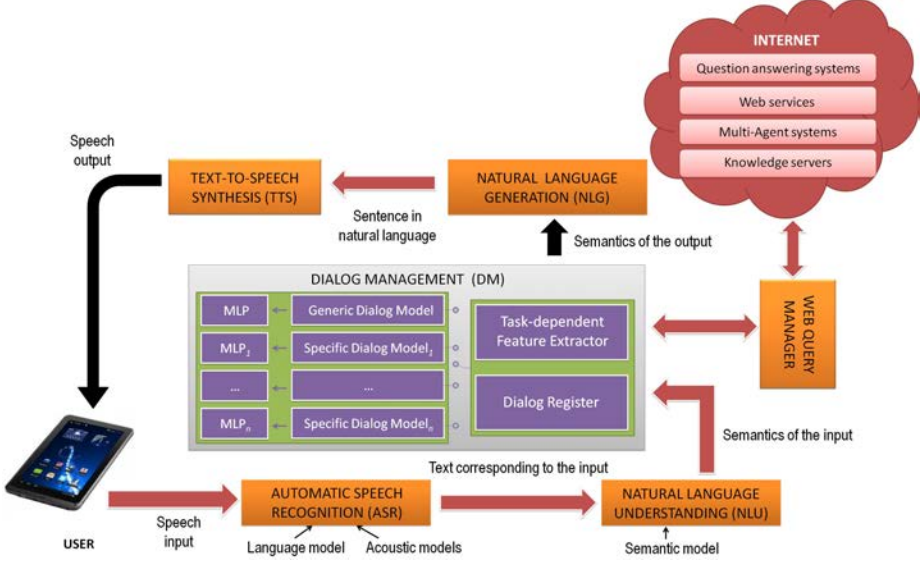


Fig. 1. Scheme of the complete architecture for the development of enhanced conversational agents

the different concepts or attributes, the confirmation of these attributes, the provision of information, and the opening and closing of the dialog. The *DR* defined for the task consists of the five possible task-dependent concepts and ten attributes previously enumerated.

Regarding the application of the proposed dialog management technique, Figure 2 shows an excerpt of a dialog for the conversational agent. Using the previously described codification for the *DR*, when a dialog starts (in the greeting turn) all the values in the dialog register are initialized to “0”. The information provided by the users in each dialog turn is employed to update the previous *DR* and obtain the current one, as Figure 2 shows.

This figure shows the semantic interpretation and confidence scores (in brackets) for a user’s utterance provided by the SLU module. In this case, the confidence score assigned to the attribute *Date* is very low. Thus, a “2” value is added in the corresponding position of the DR_1 . The concept (*Hour*) and the attribute *Destination* are recognized with a high confidence score, adding a “1” value in the corresponding positions of the DR_1 . As the input to the MLP is generated using DR_1 , the codification of the labeling of the last system turn (A_1), and the task-independent information provided in the last user turn (none in this case), the dialog manager selects to confirm the departure date. This process is repeated to predict the next system response after each user turn.

The NLG module translates the semantic representations of the system dialog acts to sentences in Spanish. Our technique consists of having a set of

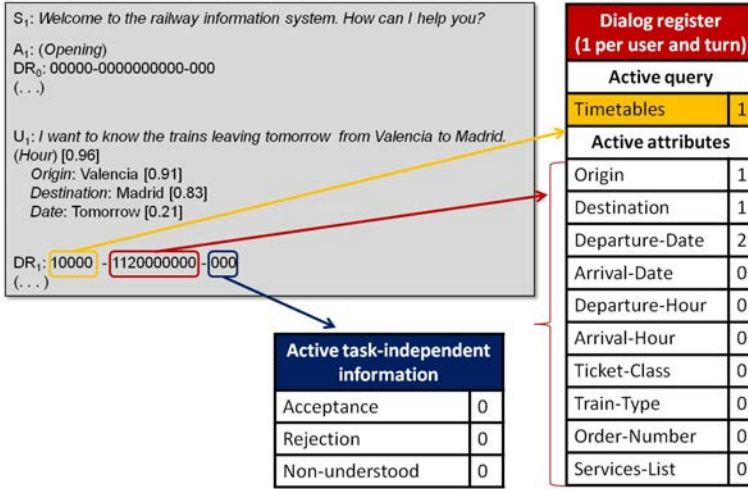


Fig. 2. Excerpt of a dialog with its correspondent *Dialog Register* and active task-independent information for one of the turns

templates associated to each one of the different dialog acts, in which the names of the attributes are shown. These names are replaced by the values recognized in order to generate the final answer for the user. For the speech output, we have integrated the Festival speech synthesis system³. The specific information relative to our task is stored in a PostGres database using information that is dynamically extracted from the web.

An initial corpus of 900 dialogs (10.8 hours) was acquired for the task by means of the Wizard of Oz technique with 225 real users [16]. A set of 20 scenarios was used to carry out the acquisition. Each scenario defined one or two objectives to be completed by the user and the set of attributes that they must provide, as Figure 3 shows. The corpus consists of 6,280 user turns, with an average number of 7.7 words per turn. Using this corpus, two versions of the conversational agent have been developed. The first one (*Conversational Agent 1*) uses a generic dialog model for the task, which has been learned using the 900 dialogs. The second one (*Conversational Agent 2*) also employs 20 specific dialog models learned using only the corresponding dialogs acquired for each one of the scenarios.

4 Results of the Evaluation

The conversational agent described in the previous section allows two operation modes. First, the system uses the ASR and the SLU modules for the normal interaction between the agent and the real users. Second, the agent allows the automatic acquisition of dialogs by means of a recently developed user-agent simulator [14]. The following subsections describes the evaluation of the two versions of the conversational agent by means of both techniques.

Code	Objective	Information provided by the user
A0	To obtain Timetables	Destination, Date
A1	To obtain Timetables	Destination, Date, Hour
A2	To obtain Timetables and Types of Trains	Destination, Date
A3	To obtain Timetables	Destination, Date, Train-Type
A4	To obtain Timetables	Destination, Date, Hour, Train-Type
A5	To obtain Timetables	Origin, Destination, Date, Hour
A6	To obtain Timetables and Types of Trains	Origin, Destination, Date
A7	To obtain Timetables	Origin, Destination, Date, Train-Type
A8	To obtain Timetables and Types of Trains	Origin, Destination, Date, Hour
A9	To obtain Timetables	Origin, Destination, Date, Hour, Train-Type
B0	To obtain Timetables and Fares	Destination, Date
B1	To obtain Timetables and Fares	Destination, Date, Hour
B2	To obtain Timetables and Fares	Destination, Date, Train-Type
B3	To obtain Timetables and Fares	Destination, Date, Hour, Train-Type
B4	To obtain Timetables and Fares	Origin, Destination, Date, Hour
B5	To obtain Timetables and Fares	Origin, Destination, Date, Train-Type
B6	To obtain Timetables and Fares	Origin, Destination, Date, Hour, Train-Type
B7	To obtain Fares	Destination, Date, Train-Type
B8	To obtain Fares	Origin, Destination, Date, Train-Type
B9	To obtain Fares	Origin, Destination, Date, Hour, Train-Type

Fig. 3. Set of different scenarios defined for the railway task

4.1 Evaluation with a User Simulator

A total of 1,000 dialogs have been acquired for each of the 20 designed scenarios by means of the interaction of the conversational agents with a brute-force automatic user simulator. The following measures were defined to compare the two corpus acquired with the conversational agents: number of successful dialogs, average number of user turns, number of different dialogs (taking into account their labeling in terms of frames and not the exact values of the attributes), the number of repetitions and user turns of the most seen dialog (in term of frames), the number of user turns of the shortest and longest dialogs, and the number of repeated dialogs (also in term of frames). Table 1 shows the result of the evaluation.

As it can be observed, the number of successfully simulated dialogs increases in most of the scenarios using the proposed technique with specific dialog models for each one of them (from a total of 6,100 successful dialogs acquired with the *Conversational Agent 1* to 8,720 successful dialogs acquired with the *Conversational Agent 2*). The user-agent simulator was developed to generate unsupervised dialogs, that is why a high amount of unsuccessful interactions were generated. In addition, there is a reduction in the average number of turns required to fulfill the objectives using the *Conversational Agent 2* (from an average of 5.7 turns using the *Conversational Agent 1* to 4.9 turns using the *Conversational Agent 2*). This general reduction in the number of turns is generalized also to the case of the longest, shortest and most seen dialogs for the *Conversational Agent 2*. Both results are specially remarkable for the most complicated subtasks, in which two objectives must be fulfilled and users must provide a large number of attributes.

On the other hand, the number of repetitions of the most seen dialog and the number of repeated dialogs is increased using the *Conversational Agent 2*. This can be explained due to the more reduced number of dialogs used to learn

Table 1. Results of the evaluation using a generic dialog model (top) or specific dialog models (bottom) and the user simulator

<i>Conversational Agent 1</i>										
	A0	A1	A2	A3	A4	A5	A6	A7	A8	A9
Number of successful dialogs	771	547	25	475	493	562	25	459	28	487
Average number of user turns	4.3	5.2	5.4	5.4	5.5	5.3	5.8	5.4	6.2	5.6
Number of different dialogs	489	500	24	438	476	493	25	437	28	466
Number of repetitions most seen dialog	70	10	2	8	5	14	1	7	1	6
Number of user turns most seen dialog	2	4	3	2	2	7	8	7	8	2
Number of user turns shortest dialog	2	2	3	2	2	2	3	2	3	2
Number of user turns longest dialog	8	8	8	8	8	8	8	8	8	8
Number of repeated dialogs	19	16	0	11	11	6	0	14	0	12
	B0	B1	B2	B3	B4	B5	B6	B7	B8	B9
Number of successful dialogs	290	215	185	167	210	184	194	247	273	263
Average number of user turns	5.6	6.2	6.2	6.2	6.3	6.0	6.3	5.6	5.6	5.8
Number of different dialogs	259	213	182	167	208	184	193	244	267	259
Number of repetitions most seen dialog	9	1	2	1	2	1	2	2	5	2
Number of user turns most seen dialog	3	3	4	4	4	8	3	3	2	5
Number of user turns shortest dialog	3	3	3	3	3	3	3	2	2	2
Number of user turns longest dialog	8	8	8	8	8	8	8	8	8	8
Number of repeated dialogs	5	1	3	3	2	3	3	3	5	5
<i>Conversational Agent 2</i>										
	A0	A1	A2	A3	A4	A5	A6	A7	A8	A9
Number of successful dialogs	831	484	96	279	353	749	403	760	391	636
Average number of user turns	3.7	4.6	5.4	3.8	4.3	3.8	5.2	4.7	4.9	4.3
Number of different dialogs	277	353	69	180	266	460	291	456	326	402
Number of repetitions most seen dialog	124	18	7	23	10	31	21	35	11	54
Number of user turns most seen dialog	2	2	4	7	3	4	8	2	3	4
Number of user turns shortest dialog	2	2	4	2	2	3	3	2	3	2
Number of user turns longest dialog	8	8	8	8	8	8	8	8	8	8
Number of repeated dialogs	14	9	1	9	5	13	2	11	2	10
	B0	B1	B2	B3	B4	B5	B6	B7	B8	B9
Number of successful dialogs	403	195	120	207	429	357	291	445	627	664
Average number of user turns	4.9	5.6	5.5	5.9	5.3	5.7	5.8	4.9	4.4	4.5
Number of different dialogs	252	178	111	187	389	336	278	318	438	547
Number of repetitions most seen dialog	51	4	4	4	6	3	4	23	19	18
Number of user turns most seen dialog	3	3	4	4	3	3	3	8	2	2
Number of user turns shortest dialog	3	3	3	3	3	3	3	2	2	2
Number of user turns longest dialog	8	8	8	8	8	8	8	8	8	8
Number of repeated dialogs	6	4	3	0	5	5	3	3	8	7

the specific dialog models, which reduces the space of dialog states in order to select the next system prompt. However, the *Conversational Agent 2* allows generating more different dialogs (from 5,552 different dialogs obtained with the *Conversational Agent 1* to 6,114 different dialogs with the *Conversational Agent 2*), then increasing the variability of the simulated corpus.

Additionally, we grouped all user and system actions into three categories: “goal directed” (actions to provide or request information), “grounding” (confirmations and negations), and “other”. Table 2 shows a comparison between these categories. As can be observed, the dialogs provided by the *Conversational Agent 2* have a better quality, as the proportion of goal-directed actions is higher than the values obtained for the *Conversational Agent 1*.

4.2 Evaluation with Real Users

Secondly, we have evaluated the conversational agents with recruited users and the same set of scenarios previously described. A total of 100 dialogs for each

Table 2. Proportions of dialog spent on-goal directed actions, ground actions and other possible actions

	<i>Conversational Agent 1</i>	<i>Conversational Agent 2</i>
Goal-directed actions	67.21%	73.43%
Grounding actions	31.64%	25.54%
Rest of actions	1.15%	1.03%

agent was recorded from the interactions of 5 users (one dialog for each scenario acquired by each user). The same set of previously described measures was used to complete an objective evaluation. As shown in Table 3, the results of the evaluation confirm the conclusions extracted from the evaluation with the user-agent simulator, obtaining a large number of successful dialogs with also a reduced number of turns when the *Conversational Agent 2* was used.

Table 3. Results of the objective evaluation with recruited users

	<i>Conversational Agent 1</i>	<i>Conversational Agent 2</i>
Number of successful dialogs	84%	92%
Average number of user turns	6.4	5.6
Number of different dialogs	76%	85%
Number of repetitions most seen dialog	5	8
Number of user turns most seen dialog	6	5
Number of user turns shortest dialog	5	3
Number of user turns longest dialog	12	10
Number of repeated dialogs	6	9

In addition, we asked the recruited users to complete a questionnaire to assess their subjective opinion about the agents performance. The questionnaire had six questions: i) Q1: *How well did the system understand you?*; ii) Q2: *How well did you understand the system messages?*; iii) Q3: *Was it easy for you to get the requested information?*; iv) Q4: *Was the interaction with the system quick enough?*; v) Q5: *If there were system errors, was it easy for you to correct them?*; vi) Q6: *In general, are you satisfied with the performance of the system?* The possible answers for each one of the questions were the same: *Never/Not at all, Seldom/In some measure, Sometimes/Acceptably, Usually/Well, and Always/Very Well*. All the answers were assigned a numeric value between one and five (in the same order as they appear in the questionnaire).

Table 4 shows the average results of the subjective evaluation using the described questionnaire. It can be observed that using either *Conversational Agent 1* or *Conversational Agent 2* the users perceived that the system understood them correctly. Moreover, they expressed a similar opinion regarding the easiness for correcting system errors. However, users said that it was easier to obtain the information specified for the different objectives using *Conversational Agent 2*, and that the interaction with the system was more adequate with the proposed dialog manager. Finally, the users were more satisfied with the system employing *Conversational Agent 2*.

Table 4. Results of the subjective evaluation with real users (For the mean value M: 1=worst, 5=best evaluation)

	<i>Conversational Agent 1</i>	<i>Conversational Agent 2</i>
Q1	M = 4.53, SD = 0.41	M = 4.71, SD = 0.33
Q2	M = 3.67, SD = 0.32	M = 3.92, SD = 0.28
Q3	M = 3.81, SD = 0.54	M = 4.29, SD = 0.32
Q4	M = 3.64, SD = 0.29	M = 4.33, SD = 0.29
Q5	M = 3.47, SD = 0.55	M = 3.54, SD = 0.53
Q6	M = 3.75, SD = 0.43	M = 4.32, SD = 0.37

5 Conclusions and Future Work

In this paper, we have adapted a statistical methodology for the development of conversational agents and the optimization of dialog strategies. The methodology is based on the estimation of a statistical model from the sequences of system and user dialog acts obtained from a set of training data. The selection of the next system response is carried out by the dialog manager using two modules. The first module is used to detect the specific objective described by the user based on the specific task-dependent semantic information provided by the SLU module. The second module is based on a classification process that takes into account the history of the dialog by means of a data structure and selects the specific dialog model generated by means of a MLP. We have defined a codification of this information to facilitate the correct operation of this classification function.

The results of the evaluation of our proposal for a conversational agent providing railway information show that the number of successful dialogs is increased in comparison with using a generic dialog model for the task. Also, these dialogs are statistically shorter and present a better quality in the selection of the system responses. For future work, we want to consider the incorporation in the *DR* of additional information regarding the user, such as specific user profiles adapted to the interaction domain. Finally, we also want to evaluate our proposal with additional domains and wider populations.

Acknowledgements. This work was supported in part by Projects MINECO TEC2012-37832-C02-01, CICYT TEC2011-28626-C02-02, CAM CONTEXTS (S2009/TIC-1485).

References

1. Pieraccini, R.: The Voice in the Machine: Building Computers that Understand Speech. The MIT Press (2012)
2. López-Cózar, R., Araki, M.: Spoken, Multilingual and Multimodal Dialogue Systems. John Wiley & Sons Publishers (2005)
3. Hempel, T.: Usability of Speech Dialog Systems: Listening to the Target Audience. Springer (2008)
4. Heinroth, T., Minker, W.: Introducing Spoken Dialogue Systems into Intelligent Environments. Kluwer Academic Publishers (2012)

5. Rodríguez, S., de Paz, Y., Bajo, J., Corchado, J.M.: Social-based Planning Model for Multi-agent Systems. *Expert Systems with Applications* 38(10), 13005–13023 (2011)
6. Corchado, J., Tapia, D., Bajo, J.: A multi-agent architecture for distributed services and applications. *Computational Intelligence* 24(2), 77–107 (2008)
7. Vipperla, R., Wolters, M., Renals, S.: Spoken dialogue interfaces for older people, pp. 118–137. IOS Press (2012)
8. Rojas-Barahona, L., Giorgino, T.: Adaptable dialog architecture and runtime engine (adarte): A framework for rapid prototyping of health dialog systems. *International Journal of Medical Informatics* 78, 56–68 (2009)
9. Schatzmann, J., Weilhammer, K., Stuttle, M., Young, S.: A Survey of Statistical User Simulation Techniques for Reinforcement-Learning of Dialogue Management Strategies. *Knowledge Engineering Review* 21(2), 97–126 (2006)
10. Griol, D., Molina, J.M., Callejas, Z.: Bringing together commercial and academic perspectives for the development of intelligent AmI interfaces. *Journal of Ambient Intelligence and Smart Environments* 4, 183–207 (2012)
11. Paek, T., Pieraccini, R.: Automating spoken dialogue management design using machine learning: An industry perspective. *Speech Communication* 50(8-9), 716–729 (2008)
12. Levin, E., Pieraccini, R., Eckert, W.: A stochastic model of human-machine interaction for learning dialog strategies. *IEEE Transactions on Speech and Audio Processing* 8(1), 11–23 (2000)
13. Young, S., Schatzmann, J., Weilhammer, K., Ye, H.: The Hidden Information State Approach to Dialogue Management. In: *Proc. ICASSP 2007*, pp. 149–152 (2007)
14. Griol, D., Carbó, J., Molina, J.: An Automatic Dialog Simulation Technique to Develop and Evaluate Interactive Conversational Agents. *Applied Artificial Intelligence* 27(9), 759–780 (2013)
15. Borrajo, M., Baruque, B., Corchado, E., Bajo, J., Corchado, J.: Hybrid neural intelligent system to predict business failure in small-to-medium-size enterprises. *International Journal of Neural Systems* 21(4), 277–296 (2011)
16. Griol, D., Hurtado, L., Segarra, E., Sanchis, E.: A Statistical Approach to Spoken Dialog Systems Design and Evaluation. *Speech Communication* 50(8-9), 666–682 (2008)
17. Minsky, M.: *A Framework for Representing Knowledge*. McGraw-Hill (1975)